

## **APPENDIX 1**

```

#!/usr/bin/php -q
<?

require("/home/scopeuser/production/include_prefs.php");
$GLOBALS["ini_config"] = $ini_config;

$path = "/var/www".$GLOBALS["ini_config"]["webpath"];
chdir($path);

require("../libs/phpmailer/class.phpmailer.php");
require("../elements/metrics_generation.inc");
require("../elements/graphing/line_plot.inc");
require("../elements/timezone.inc");

//Connect to database
mysql_connect($ini_config["hostname"],$ini_config["db_user"],$ini_config["db_password"]) or die("Unable
to connect to the database");
mysql_select_db($ini_config["database"]) or die("Unable to select database");

class Message
{
    var $elements;
    var $recipient;
    var $notification_type;
    var $timezone_offset;

    function init($elements, $recipient, $notification_type) {
        $this->elements = $elements;
        $this->timezone_offset = get_timezone_offset($this->elements["user_id"], $this-
>elements["user_server_id"]);
        $this->recipient = $recipient;
        $this->notification_type = $notification_type;
    }

    function build()
    {
        $this->elements["graphing_file"] = "line_plot";
        if ($this->notification_type == 'email') {
            $this->build_email();
        }
        elseif ($this->notification_type == 'page') {
            $this->build_page();
        }
    }

    function graph_gen() {
        $metrics_dict = Array();
        $metrics_dict["hour_from"] = 0;
        $metrics_dict["hour_to"] = 24;
        $metrics_dict["divisor"] = $this->elements["divisor"];
        $metrics_dict["legend"] = 1;
        $metrics_dict["xgrid"] = 1;
    }
}

```

```

$metrics_dict["ygrid"] = 1;
$metrics_dict["items_to_graph"] = array($this->elements["monitor_id"]);
$metrics_dict["mdr"] = 1;
$metrics_dict["watermark"] = 1;
$metrics_dict["view_type"] = "last_24";
$metrics_dict["timezone_offset"] = $this->timezone_offset * -1;
$timestamps = timestamps_last24($metrics_dict["timezone_offset"], $this-
>elements["event_log_timestamp"]);
$metrics_dict["timestamp_from"] = $timestamps[0];
$metrics_dict["timestamp_to"] = $timestamps[1];
$metrics_dict["generated_timestamps"] = generate_timestamps($metrics_dict);

$metrics_dict["plot_file"] = $this->elements["plot_type"].".php";

# Process Results Of The Last 24 Hours for the Metric
$results_data = process_results($metrics_dict, $this->elements);
$this->elements["adjusted_unit"] = $results_data[1];
# Generate Graph
$gcid = line_plot($metrics_dict, $this->elements, $results_data);
$event_log = generate_event_log($metrics_dict, $this->elements);

return Array($gcid, $event_log);
}

function build_page()
{
    // build the 120 char text body
    $page_char_max = 120;
    if ($this->elements["metric_type"] == "tunnel") {
        $dividable = 2;
        $this->page_body = $this->elements["verdict"].":".$this-
>elements["tunnel_alias"]."-" . $this->elements["tunnel_entry_alias"]."-" . $this->elements["metric_name"];
        if (sizeof($this->page_body) > $page_char_max) {
            $excess = sizeof($this->page_body - $page_char_max);
            $per_item = $excess / $dividable;
            $this->page_body = $this->elements["verdict"].":" . substr($this-
>elements["tunnel_alias"], 0, (-1 * $per_item)) . "-" . substr($this->elements["tunnel_entry_alias"], 0, (-1 *
$per_item)) . "-" . $this->elements["metric_name"];
        }
    }
    elseif ($this->elements["metric_type"] == "interface") {
        $dividable = 2;
        $this->page_body = $this->elements["verdict"].":".$this-
>elements["device_alias"]."-" . $this->elements["interface_alias"]."-" . $this->elements["metric_name"];
        if (sizeof($this->page_body) > $page_char_max) {
            $excess = sizeof($this->page_body - $page_char_max);
            $per_item = $excess / $dividable;
            $this->page_body = $this->elements["verdict"].":" . substr($this-
>elements["device_alias"], 0, (-1 * $per_item)) . "-" . substr($this->elements["interface_alias"], 0, (-1 *
$per_item)) . "-" . $this->elements["metric_name"];
        }
    }
    elseif ($this->elements["metric_type"] == "vip") {
        $dividable = 2;
        $this->page_body = $this->elements["verdict"].":".$this-
>elements["device_alias"]."-" . $this->elements["vip_alias"]."-" . $this->elements["metric_name"];
    }
}

```

```

        if (sizeof($this->page_body) > $page_char_max) {
            $exceed = sizeof($this->page_body) - $page_char_max;
            $per_item = $exceed / $dividable;
            $this->page_body = $this->elements["verdict"].":".substr($this->elements["device_alias"], 0, (-1 * $per_item))."."substr($this->elements["interface_alias"], 0, (-1 * $per_item)).".".$this->elements["metric_name"];
        }
    }
    elseif ($this->elements["metric_type"] == "device") {
        $dividable = 1;
        $this->page_body = $this->elements["verdict"].":".$this->elements["device_alias"].".".$this->elements["metric_name"];
        if (sizeof($this->page_body) > $page_char_max) {
            $excess = sizeof($this->page_body) - $page_char_max;
            $per_item = $excess / $dividable;
            $this->page_body = $this->elements["verdict"].":".substr($this->elements["device_alias"], 0, (-1 * $per_item)).".".$this->elements["metric_name"];
        }
    }
}

function build_email()
{
    // generate plot
    list($this->gcid, $this->event_log) = $this->graph_gen();

    // url for image source
    $url = $GLOBALS["ini_config"]["url"];

    // build the text and html email subjects and bodys
    if ($this->elements["metric_type"] == "tunnel") {
        $details = Array (
            Array("Tunnel Alias", $this->elements["tunnel_alias"]),
            Array("Tunnel Entry Alias", $this->elements["tunnel_entry_alias"]),
            Array("Metric", $this->elements["metric_name"]),
            Array("System Message", $this->elements["msg"])
        );
    }
    elseif ($this->elements["metric_type"] == "interface") {
        $details = Array (
            Array("Device Alias", $this->elements["device_alias"]),
            Array("Interface Alias", $this->elements["interface_alias"]),
            Array("Metric", $this->elements["metric_name"]),
            Array("System Message", $this->elements["msg"])
        );
    }
    elseif ($this->elements["metric_type"] == "vip") {
        $details = Array(
            Array("Device Alias", $this->elements["device_alias"]),
            Array("VIP Alias", $this->elements["vip_alias"]),
            Array("Metric", $this->elements["metric_name"]),
            Array("System Message", $this->elements["msg"])
        );
    }
    elseif ($this->elements["metric_type"] == "device") {
        $details = Array (

```

```

        Array("Device Alias", $this->elements["device_alias"]),
        Array("Metric", $this->elements["metric_name"]),
        Array("System Message", $this->elements["msg"])
    );
}

$formatted_details = "";

foreach ($details as $detail) {
    $formatted_details = $formatted_details.$detail[0].":      ".$detail[1]."\n";
}

$this->text_body = sprintf(
    "%s Alert Detected

%s",

    ucwords($this->elements["verdict"]),
    $formatted_details
);

if ($this->elements["verdict"] == "OK") {
    $verdict_clean = "Recovery";
}
else {
    $verdict_clean = "Error";
}

$formatted_details = '<table align="center" bgcolor="#F8F8F8">';
foreach ($details as $detail) {
    $formatted_details = $formatted_details.'<tr><td
align="right">'.$detail[0].":</td><td><b>".$detail[1]."</b></td></tr>';
}
$formatted_details = $formatted_details."</table>";
$this->html_body = sprintf(

<html>
    <head>
        <link rel="stylesheet" type="text/css" href="%sips_ccv2.css"/>
        <title>IP Sciences - Notifications - Alert</title>
    </head>
    <body bgcolor="#ffffff" marginheight=40 marginwidth=0 topmargin=40 leftmargin=0>
        <div
            align="center"
            style="border:1px solid #000; margin-height:0px; margin-left:5px; margin-
right:5px; margin-top:0px; width:100%%; height:100%%">
            <table
                style="width:100%%;height:100%%"
                summary="main layout"
                bgcolor="#ffffff"
                border="0"
                cellpadding="0" cellspacing="0">
                <tr style="height:24">
                    <td width="182" valign="top" height="24"
background="%sgraphics/hdr_orange_ban.gif">
                        
                    </td>
                    <td width="1" valign="top" height="24">
                        
</td>
<td valign="top" width="100%" bgcolor="#3c60aa" align="center"
height="24">

</td>
</tr>
<tr>
<td valign="top" bgcolor="#ffffff">

</td>
<td width="1" valign="top">

</td>
<td valign="top" width="100%">
<table width="100%" bgcolor="#888895" border="0"
cellpadding="0" cellspacing="0">
<tr>
<td bgcolor="#888895" height="17" colspan="2"></td>
</tr>
<tr>
<td bgcolor="#888895">

</td>
<td width="202" height="60" bgcolor="#888895">
</td>
</tr>
<tr>
<td colspan="2">
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td width="100%" height="100%" align="center" valign="top"
bgcolor="#ffffff" colspan="3">
<br/>
%<br/>

<br/>
'; $url, $url, $url, $url, $url, $url, $url, $url, $formatted_details);
if($this->elements['plot_type'] != 'bar_plot') {
$this->html_body .= '<table bgcolor="#000000" border="0"
cellpadding="1" cellspacing="0"><tr><td>
<table border="0" cellpadding="1" cellspacing="0"
bgcolor="#ffffff">
<tr><td colspan="3"
align="center"><h4>Legend</h4></td></tr>';
if($this->elements['metric_type'] == 'device') {
$metric_alias = $this->elements['device_alias'] . ' - ' . $this-
>elements['metric_name'];

```

```

    }
    elseif($this->elements["metric_type"] == 'interface') {
        $metric_alias = $this->elements['device_alias']. ' - ' . $this-
>elements['interface_alias']. ' - ' . $this->elements['metric_name'];
    }
    elseif($this->elements["metric_type"] == 'vip') {
        $metric_alias = $this->elements['device_alias']. ' - ' . $this-
>elements['vip_alias']. ' - ' . $this->elements['metric_name'];
    }
    elseif($this->elements["metric_type"] == 'tunnel') {
        $metric_alias = $this->elements['tunnel_alias']. $this-
>elements['tunnel_entry_alias']. ' - ' . $this->elements['metric_name'];
    }
    $metric_alias .= ' (' . $this->elements['adjusted_unit']. ')';
    $this->html_body .= '<tr>';
    $this->html_body .= '<td></td>';
    $this->html_body .= '<td style="font: 10px Arial, Sans-
Serif;">'. $metric_alias. '</td></tr>';

    //output mdr
    $this->html_body .= '<tr>';
    $this->html_body .= '<td></td>';
    $this->html_body .= '<td style="font: 10px Arial,Sans-Serif;">Median Data Range -
85%</td>';
    $this->html_body .= '</tr>';
    $this->html_body .= '<tr>';
    $this->html_body .= '<td></td>';
    $this->html_body .= '<td style="font: 10px Arial,Sans-Serif;">Median Data Range -
95%</td>';
    $this->html_body .= '</tr>';
    $this->html_body .= '<tr>';
    $this->html_body .= '<td></td>';
    $this->html_body .= '<td style="font: 10px Arial,Sans-Serif;">Median Data Range -
99%</td>';
    $this->html_body .= '</tr>';

    $this->html_body .= '</table></tr></td></table><br>';

    $this->html_body .= sprintf(
    </td></tr>');
    } else {
        $this->html_body .= sprintf('<table bgcolor="#000000" border="0"
cellpadding="0" cellspacing="0"><tr><td align="center"></td></tr></table>', $url);
    }
    $this->html_body .= '<tr><td width="100%" colspan="3">'. $this-
>event_log. "<br></td></tr>";
    $this->html_body .= sprintf(
    <tr style="height:10">
        <td align="center" width="100%" valign="top" bgcolor="#3c60aa"
colspan="3">
            
        </td>
    </tr>

```

```

        </table>
    </div>
</body>
</html>', $url);
}

function send() {
    // send an individual notification
    $mail = new PHPMailer();
    if($this->notification_type == 'email') {
        if ($this->gcid) {
            $mail-
>AddEmbeddedImage($GLOBALS['ini_config']['graphcache']. "/ips_". $this->gcid. ".png", "ipsplot", $this-
>gcid. ".png");
        }
        $mail->Subject = sprintf(
            "%s - Alert [ID:%d%s]",
            ucwords($this->elements["verdict"]),
            $this->elements["monitor_id"],
            $this->elements["account_name"]
        );
        $mail->FromName = "IP Sciences";
        $mail->From = "support@ipsciences.com";
        $mail->Body = $this->html_body;
        $mail->AltBody = $this->text_body;
    }
    else {
        $mail->FromName = "IPS";
        $mail->From = "support@ipsciences.com";
        $mail->Subject = "";
        $mail->Body = $this->page_body;
        $mail->AltBody = "";
    }

    $mail->AddAddress($this->recipient);
    $mail->Host = "localhost";
    $mail->Mailer = "smtp";

    if(!$mail->Send()) {
        echo "Message was not sent";
        echo "Mailer Error: ".$mail->ErrorInfo;
    }

    return 1;
}
}

```

```

class Notification {
    var $elements = Array();

    function init($complete_notification) {
        // initialize the elements of the notification
        $this->elements = $complete_notification;
    }
    function grab_attributes() {

```



```

// grab notification attributes
$status = $this->elements["escalation"];

$tertiary_level = (
    10 + $this->elements["primary_escalation_delay"]
    + $this->elements["secondary_escalation_delay"]
    + $this->elements["tertiary_escalation_delay"]
);

$secondary_level = (
    5 + $this->elements["primary_escalation_delay"]
    + $this->elements["secondary_escalation_delay"]
);

$primary_level = $this->elements["primary_escalation_delay"];

$this->notification_email = 0;
$this->notification_page = 0;
if ($status > $tertiary_level) {

    $this->escalation_level = "tertiary";

    if ($this->elements["tertiary_email"]) {
        $this->email_contact = $this->elements["tertiary_email"];
        $this->notification_email = 1;
    }

    if ($this->elements["tertiary_pager"]) {
        $this->pager_contact = $this->elements["tertiary_pager"];
        $this->notification_page = 1;
    }

}
if (
    ($this->notification_email == 0)
    && ($this->notification_page == 0)
    && ($status > $secondary_level)
){

    $this->escalation_level = "secondary";

    if ($this->elements["secondary_email"]) {

        $this->email_contact = $this->elements["secondary_email"];
        $this->notification_email = 1;
    }

    if ($this->elements["secondary_pager"]) {

        $this->pager_contact = $this->elements["secondary_pager"];
        $this->notification_page = 1;
    }

}
if (
    ($this->notification_email == 0)
    && ($this->notification_page == 0)
    && ($status > $primary_level)) {

```

```

        $this->escalation_level = "primary";

        if ($this->elements["primary_email"]) {
            $this->notification_email = 1;
            $this->email_contact = $this->elements["primary_email"];
        }
        if ($this->elements["primary_pager"]) {
            $this->notification_page = 1;
            $this->pager_contact = $this->elements["primary_pager"];
        }
    }
    else {
        $this->escalation_level = "none";
    }
}

function mark_sent() {
    // mark the notification sent
    $sql_mark_sent_notification = sprintf(
        "UPDATE error_log SET timestamp = timestamp, notified = 1 WHERE
error_log_id=%d AND error_log_server_id=%d",
        $this->elements['error_log_id'], $this->elements['error_log_server_id']);
    mysql_query($sql_mark_sent_notification);
}

function send() {
    // send notifications
    $error = 0;
    if ($this->notification_email) {
        $email_notification = new Message();
        $email_notification->init(
            $this->elements, $this->email_contact,
            'email');
        $email_notification->build();
        if($email_notification->send()) {
            $error = 0;
        }
    }
    if ($this->notification_page) {
        $email_notification = new Message();
        $email_notification->init(
            $this->elements, $this->pager_contact,
            'page');

        $email_notification->build();
        if($email_notification->send()) $error = 0;
    }
    if ($error == 0) return 1;
    else return 0;
}

}

function send_notifications($notification_objects) {
    // Send the notifications (one at a time in serial)

```

```

        foreach ($notification_objects as $notification_object) {
            $notification_object->grab_attributes();
            if($notification_object->send()) {
                $notification_object->mark_sent();
            }
        }
    }

function run() {
    // Run the notifications engine
    $notification_objects = Array();
    $sql_get_notifications = "SELECT DISTINCT error_log.*, DATE_FORMAT(error_log.timestamp,
'%Y%m%d%H%i%s') AS event_log_timestamp,
rule.*, rule_monitor.*, monitor.*, user.*, account.*, metric.*
FROM error_log, rule, rule_monitor, monitor, user, account, metric
WHERE error_log.notified = 0
AND rule.rule_id=error_log.rule_id
AND rule.rule_server_id=error_log.rule_server_id
AND rule_monitor.rule_id=rule.rule_id
AND rule_monitor.rule_server_id=rule.rule_server_id
AND rule_monitor.monitor_id=error_log.monitor_id
AND rule_monitor.monitor_server_id=error_log.monitor_server_id
AND monitor.monitor_id = rule_monitor.monitor_id
AND monitor.monitor_server_id = rule_monitor.monitor_server_id
AND metric.metric_id = monitor.metric_id
AND metric.metric_server_id = monitor.metric_server_id
AND user.user_id = rule.user_id
AND user.user_server_id = rule.user_server_id
AND account.account_id = user.account_id
AND account.account_server_id = user.account_server_id ORDER BY error_log.timestamp ASC";

    $queried_notifications = mysql_query($sql_get_notifications);

    while ($notification = mysql_fetch_assoc($queried_notifications)) {
        $complete_notification = $notification;
        if($complete_notification["metric_type"] == "device") {
            $sql_complete = sprintf(
                "SELECT monitor_device.*, device.* FROM monitor_device, device
WHERE monitor_device.monitor_id=%d
AND monitor_device.monitor_server_id=%d
AND device.device_id=monitor_device.device_id
AND device.device_server_id=monitor_device.device_server_id",
                $complete_notification["monitor_id"],
                $complete_notification["monitor_server_id"]);
            $complete_notification["monitor_server_id"];
        }
        elseif($complete_notification["metric_type"] == "interface") {
            $sql_complete = sprintf(
                "SELECT monitor_interface.*, interface.interface_alias,
interface.interface_number, device.*
FROM monitor_interface, interface, device
WHERE interface.interface_id=monitor_interface.interface_id
AND interface.interface_server_id=monitor_interface.interface_server_id
AND device.device_id=interface.device_id
AND device.device_server_id=interface.device_server_id
AND monitor_interface.monitor_id=%d",

```

```

AND monitor_interface.monitor_server_id=%d", $complete_notification["monitor_id"],
$complete_notification["monitor_server_id"]);
    }
    elseif($complete_notification["metric_type"] == "vip") {
        $sql_complete = sprintf(
            "SELECT monitor_vip.*, vip.vip_alias, device.*
FROM monitor_vip, vip, device
WHERE vip.vip_id = monitor_vip.vip_id
AND vip.vip_server_id = monitor_vip.vip_server_id
AND device.device_id=vip.device_id
AND device.device_server_id=vip.device_server_id
AND monitor_vip.monitor_id =%d
AND monitor_vip.monitor_server_id=%d", $complete_notification["monitor_id"],
$complete_notification["monitor_server_id"]);
    }
    elseif($complete_notification["metric_type"] == "tunnel") {
        $sql_complete = sprintf(
            "SELECT monitor_tunnel.*, tunnel.tunnel_alias,
tunnel_entry.tunnel_entry_alias
FROM monitor_tunnel, tunnel, tunnel_entry, tunnel_side
WHERE monitor_tunnel.monitor_id=%d
AND monitor_tunnel.monitor_server_id=%d
AND tunnel_side.tunnel_side_id=monitor_tunnel.tunnel_side_id
AND tunnel_side.tunnel_side_server_id=monitor_tunnel.tunnel_side_server_id
AND tunnel_entry.tunnel_entry_id=tunnel_side.tunnel_entry_id
AND tunnel_entry.tunnel_entry_server_id=tunnel_side.tunnel_entry_server_id
AND tunnel.tunnel_id=tunnel_side.tunnel_id
AND tunnel.tunnel_server_id=tunnel_side.tunnel_server_id", $complete_notification["monitor_id"],
$complete_notification["monitor_server_id"]);
    }
    $queried_notification_data = mysql_query($sql_complete);
    $more_data = mysql_fetch_assoc($queried_notification_data);
    foreach(array_keys($more_data) as $key) {
        $complete_notification[$key] = $more_data[$key];
    }
    $notification_object = new Notification();
    $notification_object->init($complete_notification);
    $notification_objects[] = $notification_object;
}

send_notifications($notification_objects);
}

run();
?>

```